

Reality Sensing, Mining and Augmentation   
for Mobile Citizen–Government Dialogue

FP7-288815

**Test scenario’s &** **results**

**C9 Server Side Mining Service  
Service Line Detection**

|  |  |
| --- | --- |
| fp7_logo | eu-flag |

co-funded by the European Union

# Template instructions

This template is used for documenting test scenarios and test results. ‘D4.4 – Technical verification and testing strategies’ describes per phase which tests need to be performed and which work package/partner is responsible for setting up and performing these tests.

Along with the software development the test scenarios are constructed based on the requirement as described in ‘D4.1 – System Architecture and Design’ and ‘D5.1 – Detailed Use Case Descriptions’.

These test scenario’s are described and agreed upon before starting the actual tests. This means that all blue sections need to be pre-filled before starting the actual test.

# Test configuration

|  |  |
| --- | --- |
| Software identification | |
| Name | C9 Server Side Mining Service – Service Line Detection |
| Versions | 0.1 |

|  |  |
| --- | --- |
| Test period | |
| Test phase | Service Level Testing |
| Test Types | Functional and performace |
| Test Status | Test plan final |
| Planned test start date | 06 Nov 2013 |
| Actual test start date | 06 Nov 2013 |
| Test completion date | End of project |
| Partners(s) | UKob |
| Tester(s) | Christoph Schaefer, Heinrich Hartmann |

|  |  |
| --- | --- |
| Test environment | |
| Test environment | None automated functional and performance test. Executed in a web browser. |
| Test devices | - |
| Test pc’s | Conventional pc, OS independent, browser independent |

|  |  |
| --- | --- |
| References | |
| Reference | Source code:  [**https://liveandgov.codespaces.com/svn/wp1/Service%20Line%20Detection**](https://liveandgov.codespaces.com/svn/wp1/Service%20Line%20Detection)  Test URL:  [**http://141.26.71.84:8080/backend/osm.html**](http://141.26.71.84:8080/backend/osm.html) |

# Test scenarios

In the table below we refer to the context mining requirements derived in D1.1.

|  |  |  |  |
| --- | --- | --- | --- |
| No. | Requirements | Expected behaviour | [OK/  NOK] |
| 1 | RB.7-13. Detect if a user is using public transportation. The individual transportation means (bus, train, tram, metro, subway, ferry) need to be distinguished. | The Service Line Detection module takes a recorded track of GPS coordinates as input and returns a list of service lines ordered by the probability the given GPS samples belonging to the respective service line.  The response time for one query should be below 100 milliseconds. | [NOK] |

# Issues raised

|  |  |
| --- | --- |
| Issue No. | SeviceLineDetectionIssue14 |
| Scenario ID | High |
| Severity | High |
| Type | Bug |
| Summary | Service Line Detection detects wrong service lines |
| Description | On the testing website (c.f. 2 References) one can select some coordinates which will be send to the server backend. The coordinates are tagged correct with the service line they belong to. As one can see in the screenshot below, the right service line (route\_id) is not detected. |
| Workaround | n/a |
| Recommendations | Integration can start independently of this issue. The accepted HTTP POST request and the response JSON is in the right format. |

|  |  |
| --- | --- |
| Issue No. | SeviceLineDetectionIssue15 |
| Severity | Low |
| Type | Bug |
| Summary | Service Line Detection response time to high |
| Description | The response time should be below 100 ms, independent how much coordinates are send to the server. |
| Workaround | Wait for the response |
| Recommendations | Integration can start; we will lower the response time before the roll out. |

# Issue screenshots

|  |  |
| --- | --- |
| Issue No. | SeviceLineDetectionIssue14 |
|  | |